**Puppy Weight Tracker**

**Describe the project in a few paragraphs**

The app will show a puppy's weight in a line graph showing how much they weighed with their age. It will allow them to add multiple puppies into the app and show all of the puppy’s weight and age, or just one puppy at once. This will allow the user to compare their puppy's weight and age, even if the puppies are not the same age at the time the records are entered. It will allow the users to enter a puppy's weight and image into the app, and store that data. The user can then see their puppy's growing progress and see what they looked like at those ages by displaying all of their images in tiles. The user will also have the option to add comments with each new entry, e.g. 30 daily walks, puppy was sick the last few days, snowy days so no walks.

**Task Vignettes (User activity "flow")**

Bob gets a new puppy by the name of Rex. He pulls up the add so can add his information to the app and begin keeping track of his growth. He pulls up the app and lands on the home page. He is presented with which pet he wants to add a new record for Dutton, Max, Add A New Pet, or All Pets.

He clicks on "Add A New Pet." He's then presented with a screen that asks him for the Pet's Name, Date of Birth, Breed, Profile Image which then creates a new record. He then types in:

* Pet’s Name: Rex
* Date of Birth: 07/12/2023
* Breed: Chihuahua
* Profile Image: Uploads an image that will be used on as the profile image which will be used on the home page and the Profile section for the pet.

Bob then clicks on “Submit.” He is now taken to a new blank page that show’s Rex’s profile information and where he can now start to add weight records for Rex. However, he notices he mistyped Rex’s Date of Birth. He hovers over Rex’s profile information and clicks on “Edit.” He updates his Date of Birth and clicks on “Submit.”

* When Bob entered Rex’s Date of Birth, this allows for automatic calculations of Rex’s age anytime a new record is added based on the Date selected in each new record.

Bob is now taken back to Rex’s page and clicks on “New” to add a new weight record for Rex. He’s now presented with a screen that asks for the Date, Weight, Upload images, and Comments. He enters all the information:

* Date: 09/10/2023.
* Weight: 8 pounds.
* Image: uploads new image.
* Comment: "Rex’s first day with us at home. He looks happy, healthy, and very playful."

He then clicks on “Submit.” The new data is all updated and present on Dutton's screen including the line graph.

Bob also needs to add a new record for Dutton, the puppy he just adopted 2 months ago. He loves him and he was just weighed today and wants to record that. The clicks on the Home icon on the upper left corner and again is presented with which puppy he wants to add a new record for Dutton, Max, Rex, Add A New Pet, or All Pets. He clicks on Dutton. This then takes him to a new screen that shows all of Dutton's profile information, and previous records that were entered. It shows a line graph with his age and weight, and some images of Dutton.

He clicks on "New" and he's then presented with a screen that asks for the Date, Weight, Upload images, and Comments. He enters all the information:

* Date: 09/10/2023.
* Weight: 32 pounds.
* Image: uploads new image.
* Comment: "Vet appointment today for vaccine. We went on daily walks most of this week."

Bob then clicks on Submit. The new data is all updated and present on Dutton's screen including the line graph.

Now with the new record uploaded, Bob wants to compare both of his dog's weight at 6 months. He clicks on the Home page button on the upper left corner, and he's back at the home page where he can select which pet he wants to see. He clicks on "All Pets."

From here, he can see a line graph that shows both Dutton's, Rex’s, and Max's weights recorded, and see all of their images based on each puppy.

Details/ideas for later:

* Can images be stored in a csv file? How will the images be stored?
* Do I want to give the user to see the weight in pounds or kilograms?
* Should I add separate graphs on individual pet pages? One that shows weight/date and another that shows weight/age?
* Should the user be able to add new entry records under the All Pets page?

**Final (self) assessment:**

The biggest unexpected change from my sketch was having to go back to the drawing board and reevaluate what to do my project on. My confidence on implementing this project seems low to medium. It’s been time consuming to have to learn everything including Python to Visual Studio Code, watching refresher videos, and everything else in between. My biggest potential problem that I’ll need to solve will be to get my code to work as expected. I am hoping this won’t be as difficult as I’m thinking after writing the technical flow.

**Technical “flow”**

* Query will return a csv file, Pets.csv, which returns every pet in the file. Each has a unique ID.
* User adds a new pet, which adds a record to the Pets.csv file with a new unique ID. This includes the pet’s name, date of birth, breed, and a profile image.
* Query returns the pet’s unique data stored in a separate csv, PetRecords.csv which stores the pet’s weight, date for the input, comments, and an image. The weight and age is shown in a line graph.
* Query on the same page return’s the information unique to that pet from the Pets.csv file.
* If user chooses, they can see all pet’s data on one screen on a line graph, instead of individual pet’s which is All Records.
* The age will be calculated based on the time for each record was added and the date of birth. Both dates coming from two separate files.
* Version 2 will be implemented as a web app using Pandas, Plotly, Dash, and HTML.

from dash import Dash, html, dcc, callback, Output, Input

import plotly.express as px

import pandas as pd

#Import Pets.csv and PetRecord.csv files

Import csv

# read file

def load\_file(fileName):

    petrecord = open(fileName)

    Reader = csv.reader(petrecord)

    For loop that will read through csv files.

    Returns file as an array.

#Adds new row to file for new pet weight record

def add\_new\_record(newPetObject, file):

    Information is entered by the user via keyboard and stores the new data in the csv file.

    For loop that will read through csv files and appends a new row.

#Adds a new pet to file.

def add\_new\_pet(PetID, file)

    Information is entered by the user via keyboard and stores the new data in the csv file.

    For loop that will read through csv files and appends a new row.

# Calculates age

def getAge():

    Calculate the age based on all of the Date fields from the PetRecord.csv file and compares to the DOB field from the pets.csv file.

#Plots csv file into plotly

 def plotcsv()

    Plots pet csv file based on PetID selected by user.

# Get Pet

def display\_UI()

    Print values per the unique PetIDs to generate buttons.

# Returns all unique pets.

def Get\_All\_Unique\_Pets(PetData)

    Loop through file and finds the uniqueID and adds to an array. Returns the array of IDs.

# Returns all unique pet's info per petID (petData).

def Get\_All\_Info\_For\_Pet(PetID, PetData)

    Returns array with all info pertaining to petID (Parsed specific to the unique PetID.)

#Returns every pet's record

def Get\_All\_Info\_For\_All\_Pets()

    Call Get\_All\_Unique\_Pets() function

    Call Get\_All\_Info\_For\_Pet() function

#Prints out Pet's Records

def print\_record\_for\_individual(PetID)

    Call Get\_All\_Info\_For\_Pet(PetID, PetData)

    Prints pet profile based on PetID selected by user from csv.

#Allow updates to Profile.

def updatedogprofile(PetID, PetProfile)

    returns updated pet's profile

# gets dog's profile based on PetID.

def get\_dogprofile(PetID, PetProfile)

    returns dog's profiles based on petID.

